Amendment to the Specification:

Please amend the specification as follows:

Please replace the pending title with the following new title:

--METHODS FOR PRODUCING ALPHA-SUBSTITUTED CARBOXYLIC ACIDS USING NITRILASES AND STRECKER REAGENTS--

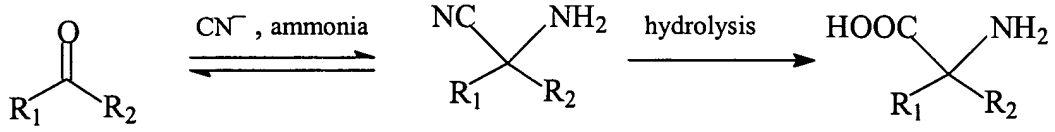
Please replace the abstract with the following rewritten paragraph:

In accordance with the present invention there are provided methods for producing enantiomerically pure a-substituted carboxylic acids, such as, for example, a-amino acids and a-hydroxy acids, said method comprising combining an aldehyde or ketone with a cyanide and ammonia or an ammonium salt or an amine, in the presence of a nitrilase or a polypeptide having nitrilase activity which stereoselectively hydrolyzes the amino nitrile or cyanohydrin intermediate under conditions sufficient to produce the enantiomerically pure α-substituted carboxylic acid, such as, for example, α-amino acids and α-hydroxy acids.

C1

Please replace the specification as follows:

On page 2, replace the first paragraph, lines 1 to 5, with the following rewritten paragraph, replacing the illustration of the chemical reaction with the following:



aldehyde or ketone

C2

On page 54, replace the paragraph spanning lines 19 to 26, with the following rewritten paragraph:

Another method for generating variants is assembly PCR. Assembly PCR involves the assembly of a PCR product from a mixture of small DNA fragments. A large number of different PCR reactions occur in parallel in the same vial, with the products of one reaction priming the products of another reaction. Assembly PCR is described in U.S. Patent No. 5,965,408, issued October 12, 1999, ~~pending~~ U.S. Patent Application Serial No. 08/677,112

C3

C3 filed July 9, 1996, entitled, Method of "DNA Shuffling with Polynucleotides Produced by Blocking or interrupting a Synthesis or Amplification Process", the disclosure of which is incorporated herein by reference in its entirety.

On page 56, replace the paragraph spanning lines 21 to 29, with the following rewritten paragraph:

In some embodiments, the variants are created using shuffling procedures wherein portions of a plurality of nucleic acids which encode distinct polypeptides are fused together to create chimeric nucleic acid sequences which encode chimeric polypeptides as described in U.S. Patent No. 5,965,408, issued October 12, 1999, ~~pending~~ U.S. Patent Application Serial No. C4 08/677,112 filed July 9, 1996, entitled, "Method of DNA Shuffling with Polynucleotides Produced by Blocking or interrupting a Synthesis or Amplification Process", and U.S. Patent No. 5,939,250, issued August 17, 1999, ~~pending~~ U.S. Patent Application Serial No. 08/651,568 filed May 22, 1996, entitled, "Combinatorial Enzyme Development".

On pages 63 to 64, replace the paragraph spanning page 63, line 22, to page 64, line 2, with the following rewritten paragraph:

Embodiments of the invention include systems (e.g., internet based systems), particularly computer systems which store and manipulate the sequence information described herein. ~~One example of a computer system 100 is illustrated in block diagram form.~~ As used herein, "a computer system" refers to the hardware components, software components, and data C5 storage components used to analyze a nucleotide sequence of a nucleic acid sequence as set forth in SEQ ID Nos: 1 and 3, and sequences substantially identical thereto, or a polypeptide sequence as set forth in SEQ ID Nos: 2 and 4. The computer system ~~100~~ typically includes a processor for processing, accessing and manipulating the sequence data. The processor ~~105~~ can be any well-known type of central processing unit, such as, for example, the Pentium III from Intel Corporation, or similar processor from Sun, Motorola, Compaq, AMD or International Business Machines.

On page 64, replace the paragraph spanning lines 4 to 8, with the following rewritten paragraph:

C6

Typically the computer system ~~100~~ is a general purpose system that comprises the processor ~~105~~ and one or more internal data storage components ~~110~~ for storing data, and one or more data retrieving devices for retrieving the data stored on the data storage components. A skilled artisan can readily appreciate that any one of the currently available computer systems are suitable.

On page 64, replace the paragraph spanning lines 10 to 15, with the following rewritten paragraph:

C7

In one particular embodiment, the computer system ~~100~~ includes a processor ~~105~~ connected to a bus which is connected to a main memory ~~115~~ (preferably implemented as RAM) and one or more internal data storage devices ~~110~~, such as a hard drive and/or other computer readable media having data recorded thereon. In some embodiments, the computer system ~~100~~ further includes one or more data retrieving device ~~118~~ for reading the data stored on the internal data storage devices ~~110~~.

On page 64, replace the paragraph spanning lines 17 to 24, with the following rewritten paragraph:

C8

The data retrieving device ~~118~~ may represent, for example, a floppy disk drive, a compact disk drive, a magnetic tape drive, or a modem capable of connection to a remote data storage system (e.g., via the internet) etc. In some embodiments, the internal data storage device ~~110~~ is a removable computer readable medium such as a floppy disk, a compact disk, a magnetic tape, etc. containing control logic and/or data recorded thereon. The computer system ~~100~~ may advantageously include or be programmed by appropriate software for reading the control logic and/or the data from the data storage component once inserted in the data retrieving device.

On page 64, replace the paragraph spanning lines 26 to 29, with the following rewritten paragraph:

C9 The computer system ~~100~~ includes a display ~~120~~ which is used to display output to a computer user. It should also be noted that the computer system ~~100~~ can be linked to other computer systems ~~125a-c~~ in a network or wide area network to provide centralized access to the computer system ~~100~~.

On page 65, replace the paragraph spanning lines 1 to 5, with the following rewritten paragraph:

C10 Software for accessing and processing the nucleotide sequences of a nucleic acid sequence as set forth in SEQ ID Nos: 1 and 3, and sequences substantially identical thereto, or a polypeptide sequence as set forth in SEQ ID Nos: 2 and 4, and sequences substantially identical thereto, (such as search tools, compare tools, and modeling tools etc.) may reside in main memory ~~115~~ during execution.

On page 65, replace the paragraph spanning lines 7 to 29, with the following rewritten paragraph:

C11 In some embodiments, the computer system ~~100~~ may further comprise a sequence comparison algorithm for comparing a nucleic acid sequence as set forth in SEQ ID Nos: 1 and 3, and sequences substantially identical thereto, or a polypeptide sequence as set forth in SEQ ID Nos: 2 and 4, and sequences substantially identical thereto, stored on a computer readable medium to a reference nucleotide or polypeptide sequence(s) stored on a computer readable medium. A "sequence comparison algorithm" refers to one or more programs which are implemented (locally or remotely) on the computer system ~~100~~ to compare a nucleotide sequence with other nucleotide sequences and/or compounds stored within a data storage means. For example, the sequence comparison algorithm may compare the nucleotide sequences of a nucleic acid sequence as set forth in SEQ ID Nos: 1 and 3, and sequences substantially identical thereto, or a polypeptide sequence as set forth in SEQ ID Nos: 2 and 4, and sequences substantially identical thereto, stored on a computer readable medium to reference sequences stored on a computer readable medium to identify homologies or structural motifs. Various sequence comparison programs identified elsewhere in this patent specification are particularly contemplated for use in this aspect of the invention. Protein and/or nucleic acid sequence

homologies may be evaluated using any of the variety of sequence comparison algorithms and programs known in the art. Such algorithms and programs include, but are by no means limited to, TBLASTN, BLASTP, FASTA, TFASTA, and CLUSTALW (Pearson and Lipman, Proc. Natl. Acad. Sci. USA 85(8):2444-2448, 1988; Altschul et al., J. Mol. Biol. 215(3):403-410, 1990; Thompson et al., Nucleic Acids Res. 22(2):4673-4680, 1994; Higgins et al., Methods Enzymol. 266:383-402, 1996; Altschul et al., J. Mol. Biol. 215(3):403-410, 1990; Altschul et al., Nature Genetics 3:266-272, 1993).

On pages 66 to 68, replace the paragraph spanning page 66, line 19, to page 68, line 21, with the following rewritten paragraph:

A "comparison window", as used herein, includes reference to a segment of any one of the number of contiguous positions selected from the group consisting of from 20 to 600, usually about 50 to about 200, more usually about 100 to about 150 in which a sequence may be compared to a reference sequence of the same number of contiguous positions after the two sequences are optimally aligned. Methods of alignment of sequence for comparison are well-known in the art. Optimal alignment of sequences for comparison can be conducted, e.g., by the local homology algorithm of Smith & Waterman, Adv. Appl. Math. 2:482, 1981, by the homology alignment algorithm of Needleman & Wunsch, J. Mol. Biol 48:443, 1970, by the search for similarity method of person & Lipman, Proc. Nat'l. Acad. Sci. USA 85:2444, 1988, by computerized implementations of these algorithms (GAP, BESTFIT, FASTA, and TFASTA in the Wisconsin Genetics Software Package, Genetics Computer Group, 575 Science Dr., Madison, WI), or by manual alignment and visual inspection. Other algorithms for determining homology or identity include, for example, in addition to a BLAST program (Basic Local Alignment Search Tool at the National Center for Biological Information), ALIGN, AMAS (Analysis of Multiply Aligned Sequences), AMPS (Protein Multiple Sequence Alignment), ASSET (Aligned Segment Statistical Evaluation Tool), BANDS, BESTSCOR, BIOSCAN (Biological Sequence Comparative Analysis Node), BLIMPS (BLocks IMProved Searcher), FASTA, Intervals & Points, BMB, CLUSTAL V, CLUSTAL W, CONSENSUS, LCONSENSUS, WCONSENSUS, Smith-Waterman algorithm, DARWIN, Las Vegas algorithm, FNAT (Forced Nucleotide Alignment Tool), Framealign, Framesearch, DYNAMIC, FILTER,

FSAP (Fristensky Sequence Analysis Package), GAP (Global Alignment Program), GENAL, GIBBS, GenQuest, ISSC (Sensitive Sequence Comparison), LALIGN (Local Sequence Alignment), LCP (Local Content Program), MACAW (Multiple Alignment Construction & Analysis Workbench), MAP (Multiple Alignment Program), MBLKP, MBLKN, PIMA (Pattern-Induced Multi-sequence Alignment), SAGA (Sequence Alignment by Genetic Algorithm) and WHAT-IF. Such alignment programs can also be used to screen genome databases to identify polynucleotide sequences having substantially identical sequences. A number of genome databases are available, for example, a substantial portion of the human genome is available as part of the Human Genome Sequencing Project ~~(J. Roach, http://weber.u.Washington.edu/~roach/human_genome_progress2.html) (Gibbs, 1995)~~. At least twenty-one other genomes have already been sequenced, including, for example, M. genitalium (Fraser et al., 1995), M. jannaschii (Bult et al., 1996), H. influenzae (Fleischmann et al., 1995), E. coli (Blattner et al., 1997), and yeast (S. cerevisiae) (Mewes et al., 1997), and D. melanogaster (Adams et al., 2000). Significant progress has also been made in sequencing the genomes of model organism, such as mouse, C. elegans, and Arabadopsis sp. Several databases containing genomic information annotated with some functional information are maintained by different organization, and are accessible via the internet~~, for example, http://wwwtigr.org/tdb; http://www.genetics.wisc.edu; http://genome-www.stanford.edu/~ball; http://hiv-web.lanl.gov; http://www.ncbi.nlm.nih.gov; http://www.ebi.ac.uk; http://Pasteur.fr/other/biology; and http://www.genome.wi.mit.edu~~. One example of a useful algorithm is BLAST and BLAST 2.0 algorithms, which are described in Altschul et al., Nuc. Acids Res. 25:3389-3402, 1977, and Altschul et al., J. Mol. Biol. 215:403-410, 1990, respectively. Software for performing BLAST analyses is publicly available through the National Center for Biotechnology Information ~~(http://www.ncbi.nlm.nih.gov/)~~. This algorithm involves first identifying high scoring sequence pairs (HSPs) by identifying short words of length W in the query sequence, which either match or satisfy some positive-valued threshold score T when aligned with a word of the same length in a database sequence. T is referred to as the neighborhood word score threshold (Altschul et al., supra). These initial neighborhood word hits act as seeds for initiating searches to find longer HSPs containing them. The word hits are extended in both directions along each sequence for as far as the cumulative alignment score can be increased. Cumulative scores are calculated using,

for nucleotide sequences, the parameters M (reward score for a pair of matching residues; always >0). For amino acid sequences, a scoring matrix is used to calculate the cumulative score. Extension of the word hits in each direction are halted when: the cumulative alignment score falls off by the quantity X from its maximum achieved value; the cumulative score goes to zero or below, due to the accumulation of one or more negative-scoring residue alignments; or the end of either sequence is reached. The BLAST algorithm parameters W, T, and X determine the sensitivity and speed of the alignment. The BLASTN program (for nucleotide sequences) uses as defaults a wordlength (W) of 11, an expectation (E) of 10, M=5, N=-4 and a comparison of both strands. For amino acid sequences, the BLASTP program uses as defaults a wordlength of 3, and expectations (E) of 10, and the BLOSUM62 scoring matrix (see Henikoff & Henikoff, Proc. Natl. Acad. Sci. USA 89:10915, 1989) alignments (B) of 50, expectation (E) of 10, M=5, N= -4, and a comparison of both strands.

C12

On page 69, replace the paragraph spanning lines 15 to 25, with the following rewritten paragraph:

The BLAST programs identify homologous sequences by identifying similar segments, which are referred to herein as "high-scoring segment pairs," between a query amino or nucleic acid sequence and a test sequence which is preferably obtained from a protein or nucleic acid sequence database. High-scoring segment pairs are preferably identified (i.e., aligned) by means of a scoring matrix, many of which are known in the art. Preferably, the scoring matrix used is the BLOSUM62 matrix (Gonnet et al., Science 256:1443-1445, 1992; Henikoff and Henikoff, Proteins 17:49-61, 1993). Less preferably, the PAM or PAM250 matrices may also be used (see, e.g., Schwartz and Dayhoff, eds., 1978, Matrices for Detecting Distance Relationships: Atlas of Protein Sequence and Structure, Washington: National Biomedical Research Foundation). BLAST programs are accessible through the U.S. National Library of Medicine~~, e.g., at www.ncbi.nlm.nih.gov~~.

C13

On page 70, replace the paragraph spanning lines 1 to 2, with the following rewritten paragraph:

C14 The database of sequences can be a private database stored within the computer system ~~100~~, or a public database such as GENBANK that is available through the Internet.

On page 70, replace the paragraph spanning lines 4 to 7, with the following rewritten paragraph:

C15 The process ~~200~~ begins at a start state ~~201~~ and then moves to a state ~~202~~ wherein the new sequence to be compared is stored to a memory in a computer system ~~100~~. As discussed above, the memory could be any type of memory, including RAM or an internal storage device.

On page 70, replace the paragraph spanning lines 9 to 19, with the following rewritten paragraph:

The process ~~200~~ then moves to a state ~~204~~ wherein a database of sequences is opened for analysis and comparison. The process ~~200~~ then moves to a state ~~206~~ wherein the first sequence stored in the database is read into a memory on the computer. A comparison is then performed at a state ~~210~~ to determine if the first sequence is the same as the second sequence. It is important to note that this step is not limited to performing an exact comparison between the

C16 new sequence and the first sequence in the database. Well-known methods are known to those of skill in the art for comparing two nucleotide or protein sequences, even if they are not identical. For example, gaps can be introduced into one sequence in order to raise the homology level between the two tested sequences. The parameters that control whether gaps or other features are introduced into a sequence during comparison are normally entered by the user of the computer system.

On page 70, replace the paragraph spanning lines 21 to 25, with the following rewritten paragraph:

Once a comparison of the two sequences has been performed at the state ~~210~~, a determination is made at a decision state ~~210~~ whether the two sequences are the same. Of course, the term "same" is not limited to sequences that are absolutely identical. Sequences that

C17 are within the homology parameters entered by the user will be marked as "same" in the process ~~200~~.

On pages 70 to 71, replace the paragraph spanning page 70, line 21, to page 71, line 5, with the following rewritten paragraph:

If a determination is made that the two sequences are the same, the process ~~200~~ moves to a state ~~214~~ wherein the name of the sequence from the database is displayed to the user. This state notifies the user that the sequence with the displayed name fulfills the homology constraints that were entered. Once the name of the stored sequence is displayed to the user, the process ~~200~~ moves to a decision state ~~218~~ wherein a determination is made whether more sequences exist in the database. If no more sequences exist in the database, then the process ~~200~~ terminates at an end state ~~220~~. However, if more sequences do exist in the database, then the process ~~200~~ moves to a state ~~224~~ wherein a pointer is moved to the next sequence in the database so that it can be compared to the new sequence. In this manner, the new sequence is aligned and compared with every sequence in the database.

C18

On page 71, replace the paragraph spanning lines 7 to 10, with the following rewritten paragraph:

It should be noted that if a determination had been made at the decision state ~~212~~ that the sequences were not homologous, then the process ~~200~~ would move immediately to the decision state ~~218~~ in order to determine if any other sequences were available in the database for comparison.

C19

On page 72, replace the paragraph spanning lines 16 to 23, with the following rewritten paragraph:

The process ~~250~~ begins at a start state ~~252~~ and then moves to a state ~~254~~ wherein a first sequence to be compared is stored to a memory. The second sequence to be compared is then stored to a memory at a state ~~256~~. The process ~~250~~ then moves to a state ~~260~~ wherein the first character in the first sequence is read and then to a state ~~262~~ wherein the first character of the second sequence is read. It should be understood that if the sequence is a nucleotide sequence, then the character would normally be either A, T, C, G or U. If the sequence is a

C20

C20 protein sequence, then it is preferably in the single letter amino acid code so that the first and sequence sequences can be easily compared.

On page 72, replace the paragraph spanning lines 25 to 31, with the following rewritten paragraph:

A determination is then made at a decision state ~~264~~ whether the two characters are the same. If they are the same, then the process ~~250~~ moves to a state ~~268~~ wherein the next characters in the first and second sequences are read. A determination is then made whether the C21 next characters are the same. If they are, then the process ~~250~~ continues this loop until two characters are not the same. If a determination is made that the next two characters are not the same, the process ~~250~~ moves to a decision state ~~274~~ to determine whether there are any more characters either sequence to read.

On page 73, replace the paragraph spanning lines 1 to 7, with the following rewritten paragraph:

If there are not any more characters to read, then the process ~~250~~ moves to a state ~~276~~ wherein the level of homology between the first and second sequences is displayed to the C22 user. The level of homology is determined by calculating the proportion of characters between the sequences that were the same out of the total number of sequences in the first sequence. Thus, if every character in a first 100 nucleotide sequence aligned with a every character in a second sequence, the homology level would be 100%.

On pages 73 to 74, replace the paragraph spanning line 21 of page 73 to line 4 of page 74, with the following rewritten paragraph:

Accordingly, another aspect of the invention is a method for determining whether a nucleic acid sequence as set forth in SEQ ID Nos: 1 and 3, and sequences substantially C23 identical thereto, differs at one or more nucleotides from a reference nucleotide sequence comprising the steps of reading the nucleic acid code and the reference nucleotide sequence through use of a computer program which identifies differences between nucleic acid sequences and identifying differences between the nucleic acid code and the reference nucleotide sequence

with the computer program. In some embodiments, the computer program is a program which identifies single nucleotide polymorphisms. The method may be implemented by the computer systems described above ~~and the method illustrated in Figure 3~~. The method may also be performed by reading at least 2, 5, 10, 15, 20, 25, 30, or 40 or more of the nucleic acid sequences as set forth in SEQ ID Nos: 1 and 3, and sequences substantially identical thereto, and the reference nucleotide sequences through the use of the computer program and identifying differences between the nucleic acid codes and the reference nucleotide sequences with the computer program.

C 23 CONT

On page 74, replace the paragraph spanning lines 16 to 26, with the following rewritten paragraph:

The process ~~300~~ begins at a start state ~~302~~ and then moves to a state ~~304~~ wherein a first sequence that is to be checked for features is stored to a memory ~~115~~ in the computer system ~~100~~. The process ~~300~~ then moves to a state ~~306~~ wherein a database of sequence features is opened. Such a database would include a list of each feature's attributes along with the name of the feature. For example, a feature name could be "Initiation Codon" and the attribute would be "ATG". Another example would be the feature name "TAATAA Box" and the feature attribute would be "TAATAA". An example of such a database is produced by the University of Wisconsin Genetics Computer Group ~~(www.gcg.com)~~. Alternatively, the features may be structural polypeptide motifs such as alpha helices, beta sheets, or functional polypeptide motifs such as enzymatic active sites, helix-turn-helix motifs or other motifs known to those skilled in the art.

C 24

On pages 74 to 75, replace the paragraph spanning line 28 of page 74 to line 2 of page 75, with the following rewritten paragraph:

Once the database of features is opened at the state ~~306~~, the process ~~300~~ moves to a state ~~308~~ wherein the first feature is read from the database. A comparison of the attribute of the first feature with the first sequence is then made at a state ~~310~~. A determination is then made at a decision state ~~316~~ whether the attribute of the feature was found in the first sequence. If the

C 25

C25 attribute was found, then the process ~~300~~ moves to a state ~~318~~ wherein the name of the found feature is displayed to the user.

On page 75, replace the paragraph spanning lines 4 to 8, with the following rewritten paragraph:

The process ~~300~~ then moves to a decision state ~~320~~ wherein a determination is made whether move features exist in the database. If no more features do exist, then the process C26 ~~300~~ terminates at an end state ~~324~~. However, if more features do exist in the database, then the process ~~300~~ reads the next sequence feature at a state ~~326~~ and loops back to the state ~~310~~ wherein the attribute of the next feature is compared against the first sequence.

On page 75, replace the paragraph spanning lines 10 to 12, with the following rewritten paragraph:

It should be noted, that if the feature attribute is not found in the first sequence at C27 the decision state ~~316~~, the process ~~300~~ moves directly to the decision state ~~320~~ in order to determine if any more features exist in the database.

On 78, replace the paragraph spanning lines 1 to 15, with the following rewritten paragraph:

Various dilutions of the reaction mixture were made in 0.1 M sodium phosphate buffer (pH 7), ranging from no dilution to a fifty times dilution. Enzyme solutions were prepared by addition of buffer to lyophilized cell lysate preparations. These cell lysates contained nitrilase C28 which had been overexpressed in a *Pseudomonas* host. Two enzyme preparations, BD 1911 (SEQ ID NO:2, encoded by SEQ ID NO:1) and BD 1921 (SEQ ID NO:4, encoded by SEQ ID NO:3), with final protein concentrations of 10.72 and 15.56 mg/ml of solution, were used. The enzyme solutions (20 ml) were added individually to each of the phenylglycine-containing dilutions (final volume 300 ml). The samples were incubated at room temperature. After 18 hours, the reactions were sampled and run on TLC (4:1:5 1-butanol: acetic acid: water). Phenylglycine was detected in the 10, 25 and 50 times dilutions, with a faint peak appearing in the 5 times dilution, for both enzymes. After 5 days, the reactions were sampled again and

analyzed by HPLC. Nitrilase BD 1921 (SEQ ID NO:4) showed approximately 10% conversion to product in the 5 times dilution of substrate nitrilase BD 1911 (SEQ ID NO:2) converted < 5% of the substrate. Higher conversions were obtained for the lower dilutions of Strecker reaction mixture (up to 35% conversion in the 50 times dilution).

C28